Multi-robot routing under connectivity constraints

Smriti Chopra [*] **Magnus Egerstedt** [*]

[*] *Georgia Institute of Technology, Atlanta, GA 30332 USA*
(e-mail: {smriti.chopra, magnus}@ece.gatech.edu)

Abstract: This paper addresses the problem of routing multiple range constrained robots to service spatially distributed requests at specified time instants, while ensuring a connected information exchange network at all times. We discuss the feasibility aspects of such a connectivity constrained routing problem. In particular, we derive the minimum number of robots required to service such requests, and we present an algorithm for the explicit construction of the corresponding routes for every robot, with total path lengths as the optimization criteria.

Keywords: assignment problems; autonomous mobile robots; connectivity constraints; vehicle routing.

1. INTRODUCTION

We consider the problem of routing multiple robots with an associated range (sensing or communication), to service *spatio-temporal* requests, where the temporal constraint implies that each request be serviced at a specified time instant. Moreover, in order to ensure that the robots can execute the mission in a collaborative manner, the induced information exchange network must be sufficiently rich. Motivated by this fact, we require that the range-constrained network induced by the positions of the robots be connected for all times. Thus, the problem in this paper, essentially, combines two fundamental topics in robotics: multi-robot routing, and connectivity ensurance in multi-robot networks.

Multi-robot routing requires multiple robots to visit a set of spatially distributed locations with routes that optimize certain criteria. It is associated with many well known problems in combinatorial optimization, a few examples being the multiple travelling salesman problem (see Bektas (2006)), and the vehicle routing problem with its many variations (e.g. Bodin et al. (1983), Bullo et al. (2010), and Ralphs (2003)). However, to solve such problems is computationally expensive. Moreover, many applications require that requests be serviced in a synchronized and sequenced manner, thus motivating the need for *spatio-temporal* requests in lieu of spatial requests. This very topic is addressed in our previous work (see Chopra and Egerstedt (2012)), where we show that such a routing problem with *spatio-temporal* requests can be formulated as a pure assignment problem, with the resulting reduction in complexity.

In this paper, we incorporate connectivity ensurance as an extension of our previous work on multi-robot routing. In general, connectivity ensurance in multi-robot networks requires techniques for ensuring connectivity of a range constrained multi-robot network during some task execution. Such techniques include using relays dedicated towards maintain sensing or communication links (e.g. Ngyen et al. (2003), Pinkney et al. (1996), and Dixon and Frew (2009)), or using formation control stategies towards motion planning (see Pereira et al. (2003) and Kan et al. (2011)). Other methods seek connectivity at *particular* time instants only, (e.g. Ponda et al. (2011)). However, we are interested in constructing routes that maintain connetivity *for all times*, while allowing dynamic assignment between robots and *spatio-temporal* requests such that no robots exist *solely* for the task of maintaining connectivity links. Moreover, we are interested in constructing such routes with total length as an optimization criteria.

In summary, we address the feasibilty aspects of the connectivity constrained routing problem central to this paper. We derive the minimum number of robots that guarantees task completion, in that, it guarantees the *existence* of routes that service all requests at specified time instants, while ensuring a connected underlying network for all times. Moreover, inspired by the work in Spanos and Murray (2005), we explicitly construct the corresponding routes for every robot. Finally, we present some simulation results that demonstrate the routing problem.

2. A MOTIVATING EXAMPLE - THE ROBOT MUSIC WALL

Previously introduced in Chopra and Egerstedt (2012), we revisit the *Robot Music Wall*: a two-dimensional magnetic-based surface with a grid of strings in different pitches that generate sound when plucked, as illustrated in Figure 1. Distinct positions on the wall correspond to distinct sound frequencies, i.e. distinct notes of an instrument, and multiple robots with the ability to traverse the wall can reach these positions and pluck at the strings above them. In other words, a robot can effectively "play" a musical note on the wall by reaching its corresponding position and plucking the string above it.

Using this set-up, we can interpret any piece of music consisting of a series of notes to be played at specified time instants, as a series of corresponding *spatio-temporal* requests (timed positions) on the music wall. We call such a series a *Score*, which contains positions that must be reached at specified time instants. Moreover, we might even require that multiple positions are reached simultaneously, akin to a musician that plays multiple notes of an instrument simultaneously with different fingers. The connectivity criteria central to this paper alludes to the natural constraints that inhibit a musicians fingers from moving arbitrarily far apart from one another at any given time instant. By routing multiple robots to service such timed positions, we can effectively "play" the piece of music associated with them on the wall (Note that we consider a timed position "serviced" the instant it is reached by a robot, i.e. we neglect on-site servicing time).

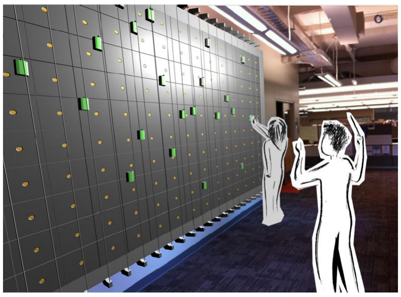

Fig. 1. A rendering of the *Robot Music Wall* concept

3. PROBLEM DEFINITION

We let $t_1, t_2, ..., t_n$ denote n discrete time instants over which the *Score* is defined, where $t_1 < ... < t_n$. Moreover, we let P_i denote the corresponding set of planar positions that require simultaneous servicing at time t_i. Each position in this set is denoted by $P_{i,\alpha}$, where $\alpha \in \{1, ..., |P_i|\}$ (the symbol $|.|$ denotes cardinality), i.e.,

$$P_i = \{P_{i,\alpha} \,|\, \alpha \in \{1, ..., |P_i|\}\}, \quad \forall i \in \{1, ..., n\} \tag{1}$$

We let $\mathcal{K}$ be the maximum number of positions that require simultaneous servicing at any time instant in T, i.e.,

$$\mathcal{K} = \max_{i \in \{1,...,n\}} |P_i| \tag{2}$$

Definition 1. Let the *Score*, denoted by Sc, be the set of all timed positions that the robots must reach. We express such timed positions as (position, time) pairs in the *Score*, i.e.,

$$Sc = \{(P_{i,\alpha}, t_i) \,|\, i \in \{1, ..., n\}, \alpha \in \{1, ..., |P_i|\}\} \tag{3}$$

Moreover, for a given set of r robots, denoted by $R = \{1, ..., r\}$, we let $P_0 = \{P_{0,\alpha} \,|\, \alpha \in \{1, ..., |P_0|\}\}$ be the set of their initial positions, defined at some initial time instant t_0. Note that for the set of robots R, we represent each robot as a planar point particle with single integrator dynamics $\dot{x}_p(t) = u_p(t)$, $p \in R$, $t \in [t_0, t_n]$. We assume that u_p is continuous almost everywhere, and we use the notation $x_p \in \hat{C}_2[t_0, t_n]$ to denote this fact (x_p denotes the differentiable almost everywhere trajectory of robot p over the interval $[t_0, t_n]$). Moreover, we let $X(t)$ denote the set of positions of all robots at time t, i.e., $X(t) = \{x_p(t) \,|\, p \in R\}$. By defining $\mathcal{C}_r$ as the following space,

$$\mathcal{C}_r = \underbrace{\hat{C}_2[t_0, t_n] \times ... \times \hat{C}_2[t_0, t_n]}_{\text{r copies}}$$

$X \in \mathcal{C}_r$ as such, denotes a collection of differentiable almost everywhere trajectories of the robots over the time interval $[t_0, t_n]$. Additionally, we let $d_{p,q}(t)$ denote the Euclidean distance between robots p and q, i.e.,

$$d_{p,q}(t) = ||x_p(t) - x_q(t)|| \tag{4}$$

Each robot has a fixed sensing range $\Delta \in \mathbb{R}$. In other words, at a given time t, robot p can sense (or "see") all robots that lie within a circle of radius Δ centered at $x_p(t)$. Since all robots possess the same range Δ, sensing links between pairs of robots are bidirectional, i.e. if robot p can sense robot q, then robot q can sense robot p as well. The positions of the robots and the resulting sensing links induce a $\Delta-$ disk proximity graph $G(X(t), \Delta)$, where the vertex set of G is given by the R, and distinct vertices p and q share an edge in G *if and only if* the distance between them $(d_{p,q})$ is at most equal to Δ, i.e.,

$$(p, q) \in E(G) \iff \Delta - d_{p,q} \geq 0 \tag{5}$$

where $E(G)$ denotes the edge set of G.

Definition 2. Given (Sc, R, P_0, Δ), $X \in \mathcal{C}_r$ is *feasible* if it satisfies the following conditions,

(a) $P_i \subseteq X(t_i) \;\forall\, i \in \{0, ..., n\}$
(b) $G(X(t), \Delta)$ is connected $\forall\, t \in [t_0, t_n]$

The first condition states that every timed position in the *Score* is reached by a robot (i.e. in context to the music wall, every note is "played"). The second condition states that the $\Delta-$disk proximity graph induced by the positions of the robots is connected for all time over the interval $[t_0, t_n]$.

We let $\mathcal{F}_r \subseteq \mathcal{C}_r$ denote the set of all *feasible* sets of trajectories i.e.,

$$\mathcal{F}_r = \{X \,|\, X \in \mathcal{C}_r \, is \, feasible\} \tag{6}$$

Thus, in order to discuss the feasibility aspects of the routing problem central to this paper, we formally define the following,

3.1 The Feasibility Problem:

Given (Sc, Δ), the objective is to find the minimum number of robots, $r^\star$ such that $\mathcal{F}_r \neq \emptyset$ for the corresponding $(Sc, R^\star, P_0, \Delta)$ quadruple, where $R^\star = \{1, ..., r^\star\}$ is the set of robots and P_0 is the set of their initial positions.

4. ESTABLISHING FEASIBILITY

In this section, we present results on the existence of a *feasible* set of trajectories.

Theorem 1. Given (Sc, R, P_0, Δ), there exists $X \in \mathcal{C}_r$ such that

(a) $P_i \subseteq X(t_i) \;\forall\, i \in \{0, ..., n\}$
(b) $G(X(t_i), \Delta)$ is connected $\forall\, i \in \{0, ..., n\}$

if and only if there exists a set of trajectories X' such that $X' \in \mathcal{F}_r$ and $X'(t_i) = X(t_i) \;\forall\, i \in \{0, ..., n\}$.

Proof. Assume that there exists $X \in \mathcal{C}_r$ that satisfies the above conditions (a) and (b). Notice that both these conditions constrain X at *only* those time instants that belong in the *Score* $(t_0, ..., t_n)$. In other words, the conditions constrain sets of robot positions $X(t_i)$, $i \in \{0, ..., n\}$.

Consider a pair of such successive sets of robot positions $X(t_{i-1})$ and $X(t_i)$. From condition (b), we see that the corresponding induced graphs $G(X(t_{i-1}), \Delta)$ and $G(X(t_i), \Delta)$ are connected. For such a pair of connected graphs, it was shown in Spanos and Murray (2005) that there exist *connectivity preserving* motions from one configuration to another. In other words, there exists $X' \in \mathcal{C}_r$ such that $X'(t_{i-1}) = X(t_{i-1})$ and $X'(t_i) = X(t_i)$, and $G(X'(t), \Delta)$ is connected over the interval (t_{i-1}, t_i). Moreover, one can see that such a X' exists between *every* pair of successive sets of positions, thereby proving the existence of a set of piecewise robot trajectories $X' \in \mathcal{F}_r$, where $X'(t_i) = X(t_i) \;\forall\, i \in \{0, ..., n\}$.

Conversely, if we assume that there exists $X'(t) \in \mathcal{F}_r$ such that $X'(t_i) = X(t_i) \;\forall\, i \in \{0, ..., n\}$, then by definition, $X(t) \in \mathcal{C}_r$ satisfies conditions (a) and (b). ∎

Theorem 1 states that the positions of the robots at *particular* time instants are sufficient to determine the existence of a *feasible* set of trajectories. However, in order to ensure that the positions satisfy conditions (a) and (b), we need to first ensure that we have enough robots. The following equations establish such a requirement for a minimum number of robots,

$$r < \mathcal{K} \Rightarrow \mathcal{F}_r = \emptyset \tag{7}$$

$$r \geq \mathcal{K} \nRightarrow \mathcal{F}_r \neq \emptyset \tag{8}$$

Equation (7) states that if the number of robots is *less than* the maximum number of positions requiring simultaneous servicing ($\mathcal{K}$) in the *Score*, then there are not enough robots to ensure that all those positions are occupied simultaneously. In other words, condition (a) would never be satisfied. As a result, there would exist no *feasible* set of trajectories ($\mathcal{F}_r = \emptyset$). Equation (8), on the other hand, states that having a minimum of $\mathcal{K}$ number of robots *still* does not guarantee the existence of a *feasible* set of trajectories. For instance, it is entirely possible that, for a given range Δ, the positions requiring simultaneous servicing at some time instant in the *Score* are located so far apart from one another that $\mathcal{K}$ number of robots are just not enough to induce a connected $\Delta-$ disk proximity graph at that time instant. In other words, condition (b) would never be satisfied, resulting in ($\mathcal{F}_r = \emptyset$).

Thus, we proceed to find the minimum number of robots $r^\star$ that ensures that conditions (a) and (b) from Theorem 1 can be met, in turn, proving the existence of a *feasible* set of trajectories. In order to keep the problem of finding $r^\star$ independent of the initial positions of the robots, we make the following assumption,

Assumption 1. The starting position of every robot in R is chosen such that the induced $\Delta-$ disk proximity graph $G(X(t_0), \Delta) = G(P_0, \Delta)$ is connected.

Theorem 2. Given a set of positions P_i specified at time t_i in the *Score*, and a sensing range Δ, the problem of finding the minimum number of robots, r_i, that ensures that every position in P_i is occupied, and the induced $\Delta-$ disk proximity graph is connected, is equivalent to the *Steiner tree problem with minimum number of Steiner points and bounded edge length (STP-MSPBEL)*

Proof. The *STP-MSPBEL* in its general form (see Lin and Xue (1999)) is stated as follows,

"Given a set of planar positions P, and a positive constant R, the objective of the STP-MSPBEL is to find a tree spanning a superset of P such that each edge in the tree has a length no more than R and the number of points other than those in P, called Steiner points, is minimized."

We see that the problem of finding the minimum number of robots at time instant t_i is identical to the *STP-MSPBEL*, where the position set P_i corresponds to P and the range Δ corresponds to the positive constant R. The positions of the vertices of the solution tree denote the positions of the robots, thus ensuring that each position in P_i is occupied, and the induced $\Delta-$ disk proximity graph is connected. ■

We denote the positions of the robots in the solution tree by S_i, and the number of *Steiner Points* by s_i. Note that $s_i + |P_i| = |S_i| = r_i$. Notice that from Theorem 2, we get the minimum number of robots required at a particular time instant in the *Score*, such that conditions (a) and (b) in Theorem 1 *evaluated at that particular time instant*, can be met. However, each time instant in the *Score* may require a different minimum number of robots, depending on its corresponding specified position set. Thus, in order to obtain a global minimum that ensures that both conditions (a) and (b) in Theorem 1 can be met, we must take the maximum over all time instants, of the minimum number of robots. We illustrate this by the following,

Theorem 3. For a given (Sc, Δ), the minimum number of robots, $r^\star$ that ensures $\mathcal{F}_{r^\star} \neq \emptyset$ for the corresponding $(Sc, R^\star, P_0, \Delta)$ quadruple, where $R^\star = \{1, ..., r^\star\}$ is the set of robots and P_0 is the set of their initial positions, is given by,

$$r^\star = \min_r \{\mathcal{F}_r \neq \emptyset\} = \max\{r_i | i \in \{1, ..., n\}\} \tag{9}$$

Proof. Let us assume that at time instant t_i, the minimum number of robots required is indeed the maximum over all time instants in the *Score*, i.e., $r_i = r^\star$. Thus, if the total number of robots r is less than $r^\star$, then at least one of the conditions (a) or (b) in Theorem 1 would never be met, thus resulting in $\mathcal{F}_r = \emptyset$.

Conversely, if the total number of robots r is at least equal to $r^\star$, then both conditions (a) or (b) in Theorem 1 can be met, thereby proving that $\mathcal{F}_r \neq \emptyset$. ■

4.1 Calculating $r^\star$

From Theorem 2, it is clear that solving the *STP-MSPBEL* is impertinent to finding $r^\star$. However, the *STP-MSPBEL* is proven to be NP-hard (see Lin and Xue (1999)). Thus, Theorems 2 and 3 provide *theoretical* results on global optimality. To calculate $r^\star$ in practical scenarios, one can use many existing algorithms with varying time complexities and performance ratios that provide an approximation to the *STP-MSPBEL* (see Chen et al. (2000) and Cheng et al. (2008)). For instance, an $\mathcal{O}(N^3)$ time approximate algorithm with performance ratio of at most 3, is presented in Cheng et al. (2008), where N denotes the number of planar positions given in the *STP-MSPBEL*. Another example is the approximate algorithm obtained by the minimum spanning tree (see Lin and Xue (1999)).

5. GENERATING TRAJECTORIES

In this section, we go from existence results to the generation of actual *feasible* trajectories. In order to achieve this, one can associate a cost with the trajectories, for example, the total length travelled. Or more precisely, for a given (Sc, R, P_0, Δ), where it is assumed that we have enough robots to ensure the existence of a set of *feasible* trajectories ($r \geq r^\star$), the objective would be to generate $X \in \mathcal{F}_r$ such that the following function is minimized,

$$\sum_{p \in R} \int_{t_0}^{t_n} \sqrt{\dot{x}_{p1}^2 + \dot{x}_{p2}^2}\, \mathrm{d}t \tag{10}$$

However, due to the high dimensionality of the multi-robot configuration space, finding a global solution i.e. a set of minimum length trajectories that are *feasible*, is typically not an option. As a result, we relax the requirement for global optimality and instead, propose the *Trajectories* algorithm that guarantees convergence to a sub-optimal solution. The main idea behind this algorithm is to apply the framework of assignment problems towards finding a solution.

In particlar, we use a key result from Chopra and Egerstedt (2012) that utilizes the *Hungarian Method* (see Kuhn (1955)) to find a so-called *optimal mapping* $A^\star : R \rightarrow 2^{Sc}$ between

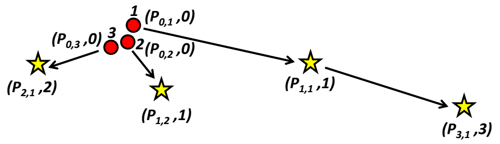

Fig. 2. An example of an *optimal mapping* between three robots (circles) and a *Score* (stars)

robots and sets of timed positions in the *Score* for a given triple (Sc, R, P_0), where an *optimal mapping* satisfies the following conditions,

(a) Every timed position in the *Score* is assigned
(b) No two robots are assigned the same timed position
(c) A robot is not assigned more than one position at a given time instant
(d) The total distance traversed by the robots (where robots move between assigned positions in straight line trajectories) is minimum

See Figure 2 for an example of an *optimal mapping* $A^\star : R \rightarrow 2^{Sc}$, where,

$$\begin{aligned}
&R = \{1, 2, 3\} \\
&Sc = \{(P_{1,1}, 1), (P_{1,2}, 1), (P_{2,1}, 2), (P_{3,1}, 3)\} \\
&A^\star(1) = \{(P_{1,1}, 1), (P_{3,1}, 3)\} \\
&A^\star(2) = \{(P_{1,2}, 1)\} \\
&A^\star(3) = \{(P_{2,1}, 2)\}
\end{aligned}$$

Moreover, the trajectory of every robot is determined by linearly interpolating between successive pairs of assigned timed positions in increasing order of specified time instants. We let $X^b \in \mathcal{C}_r$ denote the set of such trajectories. Note that X^b essentially solves the *connectivity-free* version of the routing problem central to this paper.

5.1 The Trajectories Algorithm

We proceed to explain the *Trajectories* algorithm used to generate a set of *feasible* sub-optimal trajectories that solve the connectivity constrained routing problem in this paper. The main idea behind the algorithm is as follows:

For a given (Sc, R, P_0), the *Trajectories* algorithm calculates the positions of the robots at *every* time instant in the *Score*, using the before mentioned *optimal mapping* $A^\star : R \rightarrow 2^{Sc}$. In other words, the algorithm calculates $X^b(t_i)$ for all $i \in \{1, ..., n\}$. Using these positions as an *initial* estimate, and for a given Δ, the algorithm (inspired by Theorem 1) uses the *Connect* sub-algorithm to modify these positions in a manner that ensures that the induced proximity graph at every time instant in the *Score* is connected. As a result, conditions (a) and (b) of Theorem 1 are satisfied. Moreover, keeping the optimization criteria in mind, the algorithm uses the *Assign* sub-algorithm (which essentially solves an assignment problem) to reassign robots from their positions at a particular time instant to positions at the next (successive) time instant in the *Score*. Finally, the algorithm uses the *Mid-Config* sub-algorithm to find *connectivity preserving* motions between sets of such robot positions, specified at successive time instants in the *Score*, thereby generating a set of *feasible* sub-optimal piecewise robot trajectories.

Assign (A, B, C) Given $A = \{a_1, ..., a_{|A|}\}$ specified at time instant t_{i-1}, and $B = \{b_1, ..., b_{|B|}\}$ and $C = \{c_1, ..., c_{|C|}\}$, both specified at time instant t_i, as three sets of planar positions each, $|A| \leq |B| + |C|$. Let the cost of assigning a position in A to a position in $B \cup C$ equal the distance between the two positions. The idea is to assign every position in A to a unique position in $B \cup C$, such that all positions in B are assigned, positions in C may or may not be assigned, and the total cost of assignment is minimized. In essence, the *Assign* sub-algorithm solves an unbalanced linear sum assignment problem (see Derigs (1985)).

Algorithm 1 *Trajectories* (Sc, R, P_0, Δ)

$X \leftarrow X^b$, where $X^b(t_0) = P_0$ $\{X$ is initially equal to X^b (see Chopra and Egerstedt (2012))$\}$
for $i = 1$ to n **do** $\{$iterating over all time instants in the $Score\}$
  if $G(X(t_i), \Delta)$ is connected **then**
    if $G(X^b(t_{i-1}), \Delta)$ is not connected **then** $\{$initial estimates of the robot positions at t_{i-1} required modification$\}$
      $H \leftarrow$ *Assign* $(X(t_{i-1}), X(t_i), \emptyset)$
      Using $H : X(t_{i-1}) \rightarrow X(t_i)$, update $X(t_i)$ such that the current position of robot p is given by $x_p(t_i) = H(x_p(t_{i-1})) = x^b_q(t_i)$, where $p, q \in R$ $\{$At t_i, robot p assumes the position originally occupied by robot $q\}$
      Update $X(t_j), j \in \{i+1, ..., n\}$ such that robot p assumes all positions originally occupied by robot q, at *all* future time instants in the *Score*, i.e. $x_p(t_j) = x^b_q(t_j)\ \forall j \in \{i+1, ..., n\}$
    end if
  else $\{G(X(t_i), \Delta)$ is not connected$\}$
    Find S_i, i.e. the positions of r_i robots at t_i, obtained by solving the *STP-MSPBEL* (see Theorem 2)
    if $S_i \neq P_i$ **then** $\{P_i$ does not induce a connected proximity graph, i.e. steiner points are added at $t_i\}$
      $H \leftarrow$ *Assign* $(X(t_{i-1}), S_i, X(t_i) \setminus P_i)$ $\{$e.g. see Figure 3$\}$
      Using $H : X(t_{i-1}) \rightarrow S_i \cup X(t_i) \setminus P_i$, update $X(t_i)$ such that the current position of robot p is given by $x_p(t_i) = H(x_p(t_{i-1}))$
    end if
    $X(t_i) \leftarrow$ *Connect* $(S_i, X(t_i) \setminus S_i, \Delta)$
    $H \leftarrow$ *Assign* $(X(t_{i-1}), X(t_i), \emptyset)$
    Using $H : X(t_{i-1}) \rightarrow X(t_i)$, update $X(t_i)$ such that the current position of robot p is given by $x_p(t_i) = H(x_p(t_{i-1}))$
  end if
  $X(t_{mid}) \leftarrow$ *Mid-Config* $(X(t_{i-1}), X(t_i), \Delta)$, $t_{mid} \in (t_{i-1}, t_i)$
  $X(t) \leftarrow$ linear interpolation between $X(t_{i-1})$, $X(t_{mid})$ and $X(t_i)$, $t \in (t_{i-1}, t_i)$
end for
return X

By defining $\hat{B} \triangleq \{b_1, ..., b_{|B|}, c_1, ..., c_{|C|}\} \triangleq \{\hat{b}_1, ..., \hat{b}_{|B|+|C|}\}$, we can describe the assignment problem as a linear program,

$$\min_{l} \sum_{\alpha=1}^{|A|} \sum_{\beta=1}^{|B|+|C|} ||\hat{b}_\beta - a_\alpha||\, l(\alpha, \beta) \tag{11}$$

subject to:

$$l(\alpha, \beta) \in \{0, 1\} \tag{12}$$

$$\sum_{\alpha=1}^{|A|} l(\alpha, \beta) = 1, \forall \beta \in \{1, ..., |B|\} \tag{13}$$

$$\sum_{\alpha=1}^{|A|} l(\alpha, \beta) \leq 1, \forall \beta \in \{|B|+1, ..., |B|+|C|\} \tag{14}$$

$$\sum_{\beta=1}^{|B|+|C|} l(\alpha, \beta) = 1, \forall \alpha \in \{1, ..., |A|\} \tag{15}$$

where $l(\alpha, \beta)$ represents the individual assignment of $a_\alpha \in A$ to $\hat{b}_\beta \in \hat{B}$, and is 1 if the assignment is done, and 0 otherwise.

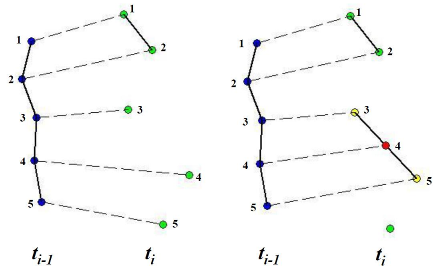

Fig. 3. Left: Robots at t_{i-1} induce a connected graph, and at t_i, a disconnected graph, Right: Using the *Assign* sub-algorithm, robots are reassigned to positions at t_i, such that all positions in S_i are occupied (yellow points represent P_i while the red point represents the added steiner point)

Note that in lines (5) and (16) of the *Trajectories* algorithm, the *Assign* sub-algorithm simply solves the balanced assignment problem for reassigning robots from their positions at a previous time instant to positions at the current time instant. A more interesting case is seen in line (12), an example of which is illustrated in Figure 3.

Algorithm 2 *Assign* (A, B, C)

```
B̂ ≜ {b_1, ..., b_|B|, c_1, ..., c_|C|} ≜ {b̂_1, ..., b̂_{|B|+|C|}}
Find l that solves Equations (11)-(15)
Find H : A → B̂ such that l(α, β) = 1 ⟺ H(a_α) = b̂_β, ∀ α ∈ {1, ..., |A|}, β ∈ {1, ..., |B| + |C|}
return H
```

Connect (A, B, Δ) Given $A = \{a_1, ..., a_{|A|}\}$ and $B = \{b_1, ..., b_{|B|}\}$, both specified at time instant t_i, as two sets of planar positions each, where the induced graph $G(A, \Delta)$ is connected, i.e. positions in A form a connected backbone, while B contains positions that may or may not be connected to this backbone. The idea is to "grow" this connected backbone by recursively adding to A, updated positions from B such that the updated $G(A, \Delta)$ becomes connected. The algorithm returns this connected backbone A. An example of such a scenario is shown in Figure 5.

Algorithm 3 *Connect* (A, B, Δ)

```
repeat
    Find α* ∈ {1, ..., |A|}, β* ∈ {1, ..., |B|} such that ||a_α* − b_β*|| = min (||a_α − b_β||) ∀α, β
    if ||a_α* − b_β*|| > Δ then
        b_β* ← a_α* + (b_β* − a_α*) / ||a_α* − b_β*|| Δ
    end if
    A ← A ∪ {b_β*}
    B ← B \ {b_β*}
until B = ∅
return A
```

Mid-Config (A, B, Δ) Given $A = \{a_1, ..., a_{|A|}\}$ specified at time instant t_{i-1}, and $B = \{b_1, ..., b_{|B|}\}$ specified at time insstant t_i, as two sets of planar positions each, where $|A| = |B|$ and the induced graphs $G(A, \Delta)$ and $G(B, \Delta)$ are both connected. The idea is to find an equal sized set of *intermediate* planar positions $M = \{m_1, ..., m_{|M|}\}$, specified at some time instant $t_{mid} \in (t_{i-1}, t_i)$, such that the induced proximity graph $G(M, \Delta)$ contains the edges of the *spanning trees* of *both* $G(A, \Delta)$ and $G(B, \Delta)$ (Notice that $G(M, \Delta)$ is connected by definition). Consequently, the set of piece-wise linear trajectories formed by linearly interpolating between A, M and B, is guaranteed to ensure a connected proximity graph for all times $t \in (t_{i-1}, t_i)$.

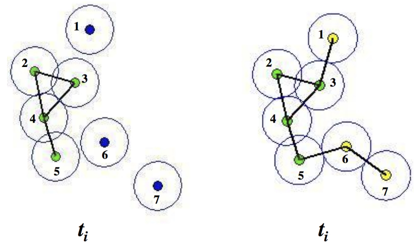

Fig. 5. Left: Robots 2,3,4 and 5 form a connected backbone while robots 1,6 and 7 are disconnected, Right: Using the *Connect* sub-algorithm, robots 1,6 and 7 merge with the connected backbone (halos around the robots depict the range $\frac{\Delta}{2}$)

Moreover, let the mid points of each unconstrained straight line path between corresonding positions a_α and b_α, $\alpha \in \{1, ..., |A|\}$ be the so-called *target* points for corresponding planar positions in M. Let $C = \{c_\alpha \,|\, c_\alpha = \frac{a_\alpha + b_\alpha}{2}, \alpha \in \{1, ..., |A|\}\}$ denote the set of these *target* points. The sub-algorithm *Mid-Config* then solves the following constrained optimization problem,

$$\min_M \sum_{\alpha=1}^{|A|} ||m_\alpha - c_\alpha|| \tag{16}$$

such that $G(M, \Delta)$ contains the edges of the spanning trees of $G(A, \Delta)$ and $G(B, \Delta)$.

Figure 6 shows an example of such a scenario.

Algorithm 4 *Mid-Config* (A, B, Δ)

```
C ≜ {c_α | c_α = (a_α + b_α)/2, α ∈ {1, ..., |A|}}
G_s(A, Δ) ← euclidean min span tree of G(A, Δ)
G_s(B, Δ) ← euclidean min span tree of G(B, Δ)
Find M by solving Equation 16 such that G(M, Δ) contains the edges of G_s(A, Δ) and G_s(B, Δ)
return M
```

Theorem 4. Given (Sc, R, P_0, Δ), where $r \geq r^\star$, the *Trajectories* algorithm finds a set of *feasible* sub-optimal trajectories that solves the connectivity constrained routing problem in this paper, with total length travelled as the optimization criteria.

Proof. As long as the number of robots is *at least* the minimum number required, the *Trajectories* algorithm updates the positions of these robots at *every* time instant in the *Score*, such that both conditions (a) and (b) of Theorem 1 are met, thereby ensuring the *existence* of a *feasible* set of trajectories. Moreover, it is able to explicitly construct such a set of trajectories, thus solving the routing problem. ■

5.2 *Optimizing Total Length: A Discussion*

In this section, we highlight the various design characteristics targetted towards optimizing the total length of robot trajectories. To begin with, the *Trajectories* algorithm uses *optimal* positions of robots, obtained by solving the *connectivity-free* version of the routing problem (see Chopra and Egerstedt (2012)), as initial estimates for finding a sub-optimal solution to the connectivity constrained routing problem. Moreover, the *Connect* sub-algorithm recursively moves each disconnected robot by a minimal distance, in order to merge it with a connected

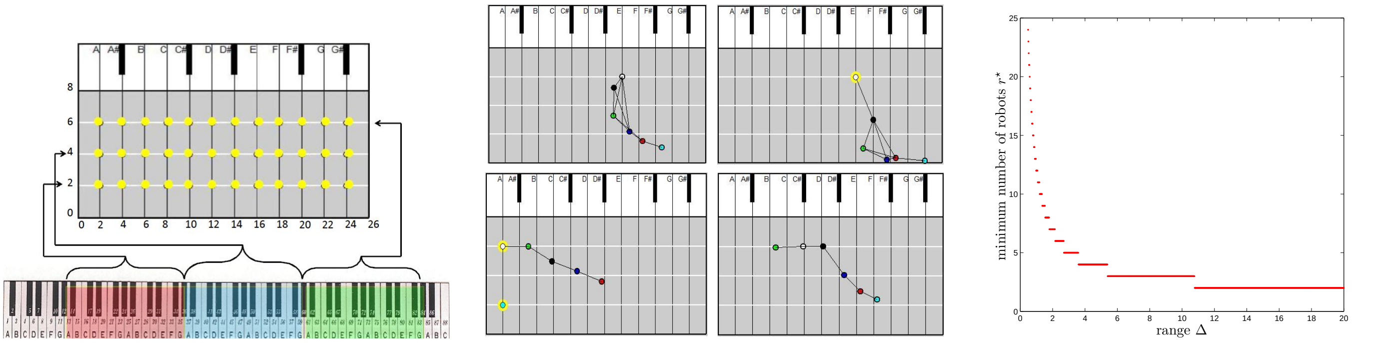

Fig. 4. A simulated *Piano Wall* with 36 coordinates (light colored points) representing the notes across three octaves of a piano (left), four snapshots of six robots playing "Für Elise", where $\Delta = 4$, $r^\star = 4$ (center), and a plot of minimum number of robots versus the range Δ

backbone at some time instant in the *Score*. The *Assign* sub-algorithm reassigns robots from their positions at one time instant to positions at the next time instant in the *Score*, such that the total length of the corresponding straight line robot trajectories between assigned positions is minimum, thereby providing a good base for the *Mid-Config* sub-algorithm. In turn, the *Mid-Config* sub-algorithm finds intermediate robot positions that cause a minimum deviation between the the original straight line trajectories and the resulting piece-wise linear ones, while satisfying the edge constraints on the induced proximity graph.

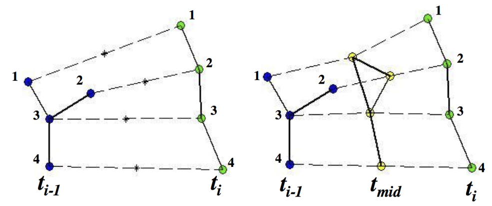

Fig. 6. Left: Robots at time instants t_{i-1} and t_i induce connected graphs, the dashed lines depict their unconstrained straight line paths, Right: Using the *Mid-Config* sub-algorithm, intermediate positions of the robots are found, and the dashed lines represent the resulting connectivity preserving piecewise linear trajectories over the interval (t_{i-1}, t_i)

6. SIMULATIONS

To demonstrate the musically inspired problem central to this paper, we simulated an example of a wall in MATLAB, instrumented to sound like a piano (see Figure 4 (left)). Our goal was to make multiple robots (simulated as 2-d points in Figure 4 (center) perform the popular composition "Für Elise" by Ludwig van Beethoven on this *Piano Wall*. Two facts about the composition "Für Elise": firstly, all notes in "Für Elise" lie amongst the set of notes used to create the *Piano Wall*, and secondly, a pianist is required to hit a maximum of two keys simultaneously throughout its performance ($\mathcal{K} = 2$).

We created the *Score* associated with "Für Elise", containing timed positions on the wall corresponding to notes in "Für Elise", specified at a beat of one second. For different values of Δ , we calculated the corresponding minimum number of robots $r^\star$ (see Figure 4 (right)). Then, for different number of robots $r \geq r^\star$ (given some Δ), we constructed the routes for every robot, using the *Trajectories* algorithm. These routes were executed by the robots with appropriate velocities that ensured their timely arrival at positions in the *Score*. In our program, the instant a robot reached an assigned timed position, it was encircled by a light circle (yellow), and the sound of the corresponding piano note was generated. Thus, our robots were able to effectively perform "Für Elise" on the *Piano Wall*. Instances of one such simulation are shown in Figure 4 (center).

REFERENCES

T. Bektas. The multiple traveling salesman problem: An overview of formulations and solution procedures. *Omega*, vol. 34, no. 3, pp. 209-219, 2006.

L. D. Bodin, B. L. Golden, A. A. Assad, and M. O. Ball. Routing and scheduling of vehicles and crews: The state of the art. *Computers and Operations Research*, vol. 10, no. 2, pp. 63–211, 1983.

F. Bullo, E. Frazzoli, M. Pavone, K. Savla, and S. L. Smith. Dynamic vehicle routing for robotic systems. *Proceedings of the IEEE*, pp. 1–23, 2011.

D. Chen, D.Z. Du, X. Hu, G. Lin, L. Wang and G. Xue. Approximations for steiner trees with minimum number of steiner points. *Journal of Global Optimization*, vol. 18, pp. 17-33, 2000.

X. Cheng, D.Z. Du, L. Wang, and B. Xu. Relay sensor placement in wireless sensor networks. *Wireless Networks*, vol. 14, no. 3, pp. 347–355, 2008.

S. Chopra, and M. Egerstedt. Multi-robot routing for servicing spatio-temporal requests: A musically inspired problem. *Analysis and Design of Hybrid Systems* to appear, 2012.

U. Derigs. The shortest augmenting path method for solving assignment problems - Motivation and computational experience. *Annals of Operations Research*, vol. 4, no. 1, pp. 57-102, 1985.

C. Dixon, and E. Frew. Maintaining optimal communication chains in robotic sensor networks using mobility control. *Mobile Networks and Applications*, vol. 14, no. 3, pp. 281–291, 2009.

Z. Kan, A. P. Dani, J. M. Shea, and W. E. Dixon Information flow based connectivity maintenance of a multi-agent system during formation control.

H. W. Kuhn. The Hungarian method for the assignment problem. *Naval Research Logistics Quarterly* vol. 2, no. 1–2, pp. 83–97, 1955.

G.H. Lin, and G.L. Xue. Steiner tree problem with minimum number of steiner points and bounded edge-length. *Information Processing Letters*, vol. 69, no. 2, pp. 53-57, 1999.

H. Nguyen, N. Pezeshkian, M. Raymond, A. Gupta, and J. Spector. Autonomous communication relays for tactical robots. *DTIC* 2003.

G. A. S. Pereira, A. K. Das, R. V. Kumar, and M. F. M. Campos. Decentralized motion planning for multiple robots subject to sensing and communication constraints. *Proceedings of the Second Multi-Robot Systems Workshop*, pp. 267–278, 2003.

M. Pinkney, D. Hampel, and S. DiPierro. Unmanned aerial vehicle (UAV) communications relay. *Military Communications Conference*, vol. 1, pp. 47–51, 1996.

S. Ponda, L. Johnson, H. Choi, and J. How. Ensuring network connectivity for decentralized planning in dynamic environments *Proceedings of the AIAA Infotech@Aerospace Conference*, 2011

T. K. Ralphs. Parallel branch and cut for capacitated vehicle routing. *Parallel Computing*, vol. 29, no. 5, pp. 607–629, 2003.

D. Spanos and R. Murray. Motion planning with wireless network constraints. *Proceedings of the American Control Conference*, vol. 1, 2005.